

REMARKS

Claims 1, 3-5, 7-18, 20-22, 24-26, and 28-38 remain pending in the present application. Independent claims 1, 13, 22, and 34 have been newly amended to more particularly recite the subject matter of the present invention. No claims have been newly added or canceled. Applicants respectfully submit that no new matter has been added. In particular, Applicants respectfully note that the application process that is a separate process from the hosting process is disclosed in the specification as filed at least in the Summary, where it is stated that the hosted runtime environment is created and loaded in a 'separate process'.

Telephone Conversation With Examiner

Applicants' representative thanks Examiner Dao for the telephone conversation conducted on January 24, 2008. Proposed claim amendments were discussed. No conclusions were reached. Applicants' representative explained the benefits of Applicants' claimed invention, and indicated that the cited art did not teach those benefits. Examiner Dao agreed to examine the application in view of the comments made during the telephone conversation.

Overview

As was previously pointed out, when developing a software application, it is known that a developer may employ a debugger to error-check or 'debug' the application. Typical software development tools allow such debugging while the application is under development merely by entering a 'debug' command. However, such typical tools suffer from an appreciable delay until the debugger and related components are loaded and the loaded debugger is attached to the process within which the application resides (*i.e.*, the application process). Moreover, such an appreciable delay is exacerbated if the debug command is repeatedly entered, and each time the delay is experienced while the debugger is reloaded and reattached to the application process.

Accordingly, in the present invention, such appreciable delays are reduced if not eliminated by loading and attaching the debugger only a single time, and prior to the first invocation of such debugger. In particular, in the present invention as now recited in the claims, the debugger is attached to a hosting process that is separate from the application process, and accordingly need only be attached to the hosting process only the single time, as may be appreciated. Thus, each time the debugger is invoked on the application, the application is loaded from the application process into the hosting process with the attached debugger already attached thereto, and debugging may then proceed without the aforementioned appreciable delay. As should be understood, such application may be loaded from the application process into the hosting process any number of times with the debugger already attached to the hosting process and without the appreciable delay.

Claim Rejections

In the Office Action, the Examiner has rejected claims 13-18, 20-22, 24-26, and 28-33 under 35 U.S.C. § 101 as being non-statutory subject matter. However, it appears from the text of the Section 101 rejection that claims 34-38 stand so rejected, and not claims 22, 24-26, and 28-33. Regardless, Applicants respectfully traverse the Section 101 rejection insofar as it may be applied to the claims as amended.

The Examiner suggests amending independent claims 13 and 34 to recite a computer-readable storage medium, and Applicants have done so. As a result, Applicants respectfully request reconsideration and withdrawal of the Section 101 rejection.

The Examiner has also rejected claims 1, 3-5, 7-18, 20-22, 24-26, and 28-38 under 35 U.S.C. § 102 as being anticipated by Bogle *et al.* (U.S. Pat. No. 6,353,923). Applicants respectfully traverse the Section 102 rejection insofar as it may be applied to the claims as amended.

Independent claim 1 as amended recites a method for debugging an application operating within a runtime environment and loaded into an application process. In the method, a hosting process is created to be separate from the application process and not based

on the application to be debugged, the runtime environment is started in the hosting process, a debugger is attached to the hosting process, and selected assemblies are preloaded into an application domain, all prior to invoking the debugger or receiving a request to debug. The attached debugger is caused to enter into a hosting process mode where the attached debugger is to debug the application as loaded from the application process into the hosting process.

Upon receiving such a request to debug the application, the application is loaded from the application process into the hosting process, and the debugger is invoked with respect to the hosting process and not the application process. Thus, and as was pointed out above, the debugger as attached to the hosting process and not the application process need not be reloaded and reattached every time debugging is requested.

Independent claims 13, 22, and 34 all recite subject matter similar to that of claim 1, albeit in the form of a system, computer-readable medium, and software development system, respectively.

The Bogle reference discloses an active debugging environment for debugging a virtual application that contains program language code from multiple compiled and/or interpreted programming languages. The active debugging environment is language neutral and host neutral, where the host is a standard content centric script host with language engines for each of the multiple compiled and/or interpreted programming languages represented in the virtual application. Notably, although the Bogle reference discloses a host, such host is not a hosting process that is separate from an application process, as is recited in claims 1, 13, 22, and 34 of the present application.

More specifically, Applicants respectfully submit that the Bogle reference fails to disclose or even suggest that an application to be debugged is loaded in an application process, that a hosting process is created to be separate from the application process and not based on the application to be debugged, that a debugger is attached to the hosting process prior to invoking the debugger or receiving a request to debug, and the attached debugger is caused to enter into a hosting process mode where the attached debugger is to debug the application as loaded from the application process into the hosting process, that upon

receiving such a request to debug the application, the application is loaded from the application process into the hosting process, and that the debugger is invoked with respect to the hosting process and not the application process, all as is recited in claims 1, 13, 22, and 34. Notably, the Bogle reference does not even appreciate that appreciable delays are experienced each time a request to debug is received because of the time needed to reload the debugger and reattach the debugger to an application process, or that such delays may be obviated by loading the debugger only a single time and attaching the debugger to a hosting process, and by loading the application from the application process to the hosting process each time a request to debug is received.

Accordingly, Applicants respectfully submit that the Bogle reference fails to disclose or even suggest all of the elements and limitations recited in claims 1, 13, 22, and 34. As a result, Applicants respectfully submit that such Bogle reference cannot be applied to anticipate such claims or any claims depending therefrom, including claims 3-5, 7-12, 14-18, 20, 21, 24-26, 28-33 and 35-38. Thus, Applicants respectfully request reconsideration and withdrawal of the Section 102 rejection.

DOCKET NO.: MSFT-1757

PATENT

Application No.: 10/626,958

Office Action Dated: November 23, 2007

In view of the foregoing Amendment and Remarks, Applicants respectfully submit that the present application including claims 1, 3-5, 7-18, 20-22, 24-26, and 28-38 is in condition for allowance and such action is respectfully requested.

Respectfully Submitted,

Date: February 21, 2008

/Joseph F. Oriti/
Joseph F. Oriti
Registration No. 47,835

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439